# Narrowband Tracking Using a Markov Random Field Algorithm

Dr. Paul M. Baggenstoss
Naval Undersea Warfare Center
Newport RI, 02841
401-832-8240 (TEL)
p.m.baggenstoss@ieee.org
This work was supported by the Office of Naval Research (ONR321US)

## I. Introduction

We present an algorithm for formulating the narrowband (or target) tracking problem as a Markov Random Field (MRF) with discrete and continous-valued hidden state variables. We then derive a simplified algorithm to estimate the model state variables. An MRF exists whenever there is a collection of sites that statistically interact with their neighbors. This requires defining a neighborhood system for determining which sites are the neighboring sites - such as a distance metric. If the Markov property is met, then if we are trying to make statistical inference about a given site, knowledge of all the sites is no better than knowledge of the state variables of only the neighboring sites.

In the narrowband tracking problem, we assume we have detected a number of "interesting sites" in a spectrogram. These "interesting sites" are regions where there appears to be straight-line motion of a narrowband signal, perhaps detected by the application of a radon transform. Each site $n$ has a measurement of center frequency, $f_n$, frequency rate $r_n$, and amplitude $a_n$. We seek to make "soft" connections between the interesting sites so as to make longer tracks. We apply a linear dynamical model, to explain the behaviour of the target. This is identical in formulation to a Kalman filter, with the exception that the state transition matrix is many-to-one (many neighbor sites to a single site). The method is generalizable to any tracking problem.

## II. Generic Algorithm

### A. Basic Notation and Goal

Let there be $L$ "interesting sites" and suppose that at each site $n$ we have a measurement $\mathbf{x}_n = [f_n, r_n, a_n]'$. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_L]$ be all of the data. Let $N_n$ be the number of neighbors of site $n$ based on some a symmetric rule for determining "neighborness". We assume that there is an underlying state at each site which we cannot observe, but can estimate. The state consists of a discrete state variable $1 \le s_n \le P$, and a continuous state variable $\boldsymbol{\theta}_n \in \mathcal{R}^p$. In the narrowband tracking problem, the discrete state $s_n$ takes on a value from 1 to 4:

1) Internal site (having both a left and right association).
2) Left terminal site (having just a right association).
3) Right terminal site (having just a left association).
4) Noise site (not associated with any other sites).

Our goal is to determine the *aposteriori* discrete probabilities $p(s_n|\mathbf{X})$ and the parameters of the conditional *aposteriori* continuous PDF $p(\boldsymbol{\theta}_n|s_n, \mathbf{X})$ at each site $n$. For tractability, we will assume $p(\boldsymbol{\theta}_n|s_n, \mathbf{X})$ is Gaussian which means we need only find the conditional means and covariances. The basis of our approach is the Markov property and the construction of the conditional probability functions (conditioned on knowing the state variables of the neighnboring sites) thus the resulting estimates are the estimates of a Markov random field (MRF). Our method does not approach the problem as a Gibbs field as does much of the MRF literature to date [1]. Our approach is iterative and convergence is achieved when $p(s_n|\mathbf{X})$ and $p(\boldsymbol{\theta}_n|s_n, \mathbf{X})$ stop changing. To start the iterative algorithm, we initialize $p(s_n|\mathbf{X})$ and $p(\boldsymbol{\theta}_n|s_n, \mathbf{X})$ to "generic" values, then design an algorithm to re-compute the same quantities at each site $n$ based on the state variables of the neighboring sites and the current site's measurement $\mathbf{x}_n$. The method is related to dependency networks [2] and iterative conditional modes [3]. The novelty of the approach is the calculation of the Markov conditional probability.

### B. Simplified Notation

Because in the discussion below, all our equations have to do with site $n$, we simplify the notation by dropping the subscript $n$ and use $i$ to represent the discrete state $s_n$ thus:

$$N \triangleq N_n,$$

$$p_n(\boldsymbol{\theta}|i, \mathbf{X}) \triangleq p(\boldsymbol{\theta}_n|s_n = i, \mathbf{X}),$$

$$Q_{n,i} \triangleq p(s_n = i|\mathbf{X}),$$

$$\mathbf{x} \triangleq \mathbf{x}_n.$$

To avoid confusion, we use the symbol $\boldsymbol{\phi}_j$ in stead of $\boldsymbol{\theta}$ to designate the continuous state variable of a neighbor site, where $1 \le j \le N$ runs over the neighbor sites of site $n$.

1-4244-0115-1/06/$20.00 ©2006 IEEE

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **01 SEP 2006** | **N/A** | **-** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **Narrowband Tracking Using a Markov Random Field Algorithm** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Undersea Warfare Center Newport RI, 02841** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release, distribution unlimited**

13. SUPPLEMENTARY NOTES
**See also ADM002006. Proceedings of the MTS/IEEE OCEANS 2006 Boston Conference and Exhibition Held in Boston, Massachusetts on September 15-21, 2006. Federal Government Rights, The original document contains color images.**

14. ABSTRACT

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | |
| **unclassified** | **unclassified** | **unclassified** | **UU** | **6** | |

### C. Single-State Assumption and Expansion over Neighbor Configurations

We assume that neighboring sites can be either associated (connected) or not. For each configuration of associated neighbor sites (by "configuration", we mean how many connected neighbors and which ones), only one state is possible at site $n$. We call this the "single state" assumption, the result of which is that as we enumerate all the possible combinations of neighbor states and "connectedness", we find that the combinations organize themselves into disconnected sets that "support" just one state at the current site.

Let the variable $K$ represent a particular association/state set. By this, we mean, $K$ represents a choice of associated (connected) neighbors as well as their states. Let $J$ represent a particular combination of states for the non-associated neighbors. The combination $\{J, K\}$ therefore specifies which neighbors are "connected" to site $n$, their states, as well as the states of all the neighbors which are not connected. Having defined $K, J$, we expand

$$p(\boldsymbol{\theta}, i|\mathbf{X}) \;=\; \sum_K \sum_{J\in\mathcal{J}_K} p(\boldsymbol{\theta}, i|K, J, \mathbf{X})\; p(K, J|\mathbf{X})$$

where $\mathcal{J}_K$ is the set of all combinations of states for the neighbors that are not connected under configuration $K$. We need not consider illegal combinations such as when a site has two "connected" right neighbors. Because the states of the non-connected neighbors does not matter, we drop the dependence on $J$ in the first term:

$$p(\boldsymbol{\theta}, i|\mathbf{X}) \;=\; \sum_K \sum_{J\in\mathcal{J}_K} p(\boldsymbol{\theta}, i|K, \mathbf{X})\; p(K, J|\mathbf{X})$$

We now divide the data into two parts:

$$\mathbf{X} = \{\mathbf{x}, \mathbf{X}^n\},$$

where $\mathbf{X}^n$ is all the data except site $n$. We expand the first term:

$$\begin{aligned} p(\boldsymbol{\theta}, i|K, \mathbf{X}) &= p(\boldsymbol{\theta}, i|\mathbf{x}, K, \mathbf{X}^n) \\ &= \tfrac{p(\boldsymbol{\theta}, i, \mathbf{x}|K, \mathbf{X}^n)}{p(\mathbf{x}|K, \mathbf{X}^n)} \\ &= \tfrac{p(\boldsymbol{\theta}, i, \mathbf{x}|K, \mathbf{X}^n)}{\sum_i \int_\theta \{\text{numerator}\} d\theta} \end{aligned}$$

Expanding the numerator and plugging in, we get

$$p(\boldsymbol{\theta}, i|\mathbf{X}) \;=\; \sum_K \sum_{J\in\mathcal{J}_K} \left\{ \tfrac{p(\mathbf{x}|\boldsymbol{\theta}, i)\; p(\boldsymbol{\theta}|i, K, \mathbf{X}^n)\; p(i|K, J, \mathbf{X}^n)}{\sum_i \int_\theta \{\text{numerator}\} d\theta} \right\}$$

$$\cdot p(K, J|\mathbf{X}),$$

Because of the single-state assumption, for a particular $K$, only one state $i$ is possible. Thus, $p(i|K, J, \mathbf{X}^n) = 1$ for $K \in \mathcal{K}_i$ where $\mathcal{K}_i$ be the set of associations that give rise to state $i$. Then

$$p(\boldsymbol{\theta}, i|\mathbf{X}) \;=\; \sum_{K\in\mathcal{K}_i} \sum_{J\in\mathcal{J}_K} \left\{ \tfrac{p(\mathbf{x}|\boldsymbol{\theta}, i)\; p(\boldsymbol{\theta}|i, K, \mathbf{X}^n)}{\int_\theta \{\text{numerator}\} d\theta} \right\}$$

$$\cdot p(K, J|\mathbf{X})$$

### D. Neighbor continuous State Variables

Let us expand the notion of neighbor configuration $K$ to include the continous state variable of the neighbors, denoted by $\boldsymbol{\phi}$. It is necessary to integrate over $\boldsymbol{\phi}$. Thus,

$$p(\boldsymbol{\theta}, i|\mathbf{X}) \;=\; \sum_{K\in\mathcal{K}_i} \sum_{J\in\mathcal{J}_K} \int_{\boldsymbol{\phi}} \left\{ \tfrac{p(\mathbf{x}|\boldsymbol{\theta}, i)\; p(\boldsymbol{\theta}|i, K, \boldsymbol{\phi}, \mathbf{X}^n)}{\int_\theta \{\text{numerator}\} d\theta} \right\}$$

$$\cdot p(K, J, \boldsymbol{\phi}|\mathbf{X})\; d\phi$$

We can simplify the above if we assume that the PDF $p(K, J, \boldsymbol{\phi}|\mathbf{X})$ is sharply defined in $\boldsymbol{\phi}$ as compared to $p(\boldsymbol{\theta}|i, K, \boldsymbol{\phi}, \mathbf{X}^n)$ for a given value of $\boldsymbol{\theta}$. This assumption is good because in the latter, $\boldsymbol{\phi}$ appears only as a conditional variable. Then, we may approximate the above as

$$p(\boldsymbol{\theta}, i|\mathbf{X}) \;=\; \sum_{K\in\mathcal{K}_i} \sum_{J\in\mathcal{J}_K} \left\{ \tfrac{p(\mathbf{x}|\boldsymbol{\theta}, i)\; p(\boldsymbol{\theta}|i, K, \boldsymbol{\phi}^*_K, \mathbf{X}^n)}{\int_\theta \{\text{numerator}\} d\theta} \right\}$$

$$\cdot p(K, J, \boldsymbol{\phi}^*_K|\mathbf{X}),$$

where $\boldsymbol{\phi}^*_K$ is the estimate of $\boldsymbol{\phi}$ corresponding to neighbor configuration $K$. This assumption is a good one because

### E. Gaussian Assumption

Assuming there is zero-mean Gaussian measurement noise, we have

$$p(\mathbf{x}|\boldsymbol{\theta}, i) = N(\mathbf{x} - \boldsymbol{\theta}, \mathbf{C}),$$

where

$$N(\mathbf{x}, \mathbf{C}) \triangleq (2\pi)^{-p/2} |\mathbf{C}|^{-1/2} \exp\{-\mathbf{x}'\mathbf{C}^{-1}\mathbf{x}\}.$$

Let

$$p(\boldsymbol{\theta}|i, K, \boldsymbol{\phi}^*_K, \mathbf{X}^n) = N(\boldsymbol{\theta} - F_i\boldsymbol{\phi}^*_K, \mathbf{F}_i'\mathbf{R}_K\mathbf{F}_i + \boldsymbol{\Sigma}_i),$$

where $\mathbf{R}_K$ is the covariance of $\boldsymbol{\phi}^*_K$ and $\mathbf{F}_i$, $\boldsymbol{\Sigma}_i$ are the continuous state transition matrix and the plant variance. With these Gaussian assumptions,

$$p(\boldsymbol{\theta}, i|\mathbf{X}) = \sum_{K\in\mathcal{K}_i} \sum_{J\in\mathcal{J}_K}$$

$$\left\{ \tfrac{N(\mathbf{x}-\boldsymbol{\theta}, \mathbf{C})\; N(\boldsymbol{\theta}-F_i\boldsymbol{\phi}^*_K, \mathbf{F}_i'\mathbf{R}_K\mathbf{F}_i+\boldsymbol{\Sigma}_i)}{\int_\theta \{\text{numerator}\} d\theta} \right\}\; p(K, J|\mathbf{X}),$$

We now drop the subscripts and superscripts from $\mathbf{F}_i,\ \boldsymbol{\Sigma}_i,\ \mathbf{R}_K,\ \boldsymbol{\phi}^*_K$ for simplicity. Using the identy

$$N(\mathbf{x} - \boldsymbol{\theta}, \mathbf{C})\; N(\boldsymbol{\theta} - F\boldsymbol{\phi}, \mathbf{F}'\mathbf{R}\mathbf{F} + \boldsymbol{\Sigma}) =$$

$$N(\mathbf{x} - F\boldsymbol{\phi}, \mathbf{F}'\mathbf{R}\mathbf{F} + \boldsymbol{\Sigma} + \mathbf{C})\; N(\boldsymbol{\theta} - \mathbf{u}_K, \mathbf{G}_K),$$

where

$$\mathbf{G}_K = \left(\mathbf{C}^{-1} + (\mathbf{F}'\mathbf{R}\mathbf{F} + \boldsymbol{\Sigma})^{-1}\right)^{-1} \tag{1}$$

$$\mathbf{u}_K = \mathbf{G}_K(\mathbf{C}^{-1}\mathbf{x} + (\mathbf{F}'\mathbf{R}\mathbf{F} + \boldsymbol{\Sigma})^{-1}\mathbf{F}\boldsymbol{\phi}), \tag{2}$$

and carrying out the integration in the denominator we get

$$p(\boldsymbol{\theta}, i|\mathbf{X}) \;=\; \sum_{K\in\mathcal{K}_i} \sum_{J\in\mathcal{J}_K} N(\boldsymbol{\theta} - \mathbf{u}_K, \mathbf{G}_K)\; p(K, J|\mathbf{X})$$

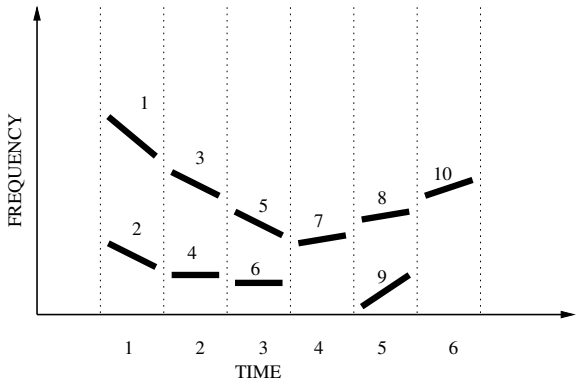


Fig. 1. Typical narrowband situation showing 8 sites.

This can be re-written as a Gaussian mixture of the form

$$p(\boldsymbol{\theta}, i|\mathbf{X}) = \left[\sum_k \alpha_k \; N(\boldsymbol{\theta} - \mathbf{u}_k, \mathbf{G}_k)\right] \; p(i|\mathbf{X}),$$

where $k$ is defined as the pair $\{K \in \mathcal{K}_i, \; J \in \mathcal{J}_K\}$, and where

$$p(i|\mathbf{X}) = \sum_{K \in \mathcal{K}_i} \sum_{J \in \mathcal{J}_K} p(K, J|\mathbf{X}), \tag{3}$$

$$\alpha_k = \frac{p(K, J|\mathbf{X})}{p(i|\mathbf{X})}.$$

Note that $\sum_k \alpha_k = 1$. To get a meaningful result, we need to collapse it down to a Gaussian PDF. The mean is:

$$\boldsymbol{\mu} = \sum_i \alpha_i \mathbf{u}_i$$

And

$$\mathcal{E}(\mathbf{x}\mathbf{x}') = \sum_i \alpha_i \left[\mathbf{G}_i + \mathbf{u}_i \mathbf{u}_i'\right]$$

So the covariance is

$$\mathcal{E}(\mathbf{x}\mathbf{x}') - \boldsymbol{\mu}\boldsymbol{\mu}' = \sum_i \alpha_i \left[\mathbf{G}_i + \mathbf{u}_i \mathbf{u}_i'\right] - \boldsymbol{\mu}\boldsymbol{\mu}'.$$

We must now determine how to get an expression for $p(K, J|\mathbf{X})$.

### *F. Narrowband Tracker Application*

Refer to Figure 1 which shows a typical narrowband (NB) situation. Each site is detected from a spectrogram and comes with a measurement of amplitude (SNR), center frequency, and frequency rate. Local frequency rate is determined by integration of the spectrogram along short lines with varying center frequency and slope, then choosing sites which are local maxima in both center frequency and slope and exceed some minimum amplitude threshold. In the figure are shown 10 sites.

In Figure 2, the neighbor sites of site 7 are shown inside the circle. We define a neighbor as any site exactly 1 frame away in time and within some minimum center frequency distance. A neighbor must be either to the right by 1 time step or to the left by one time step. We do not allow a neighbor at the same time step. All neighbors must be mutual neigbors, i.e., a site

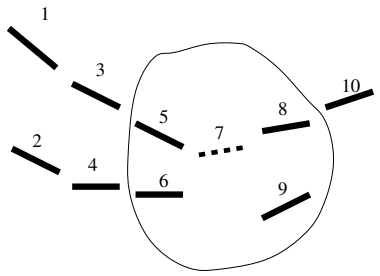


Fig. 2. Neighbors of site 7.

that has a neighbor must also be the neighbor of that site. We assume that a neighbor can be either **related** or **unrelated** to site $n$. In Figure 2, site 6 is probably associated with site 4, and not with site 7. Sites 5 and 8 are probably associated with site 7. Site 9 is probably not associated with any other sites.

We consider four possible states for a site.

1) $s_n = 1$ (Internal site). Internal sites are connected to one left neighbor (internal or left terminal site) and one right neighbor (internal or right terminal site). An internal site can have an unlimited number of unrelated sites as neigbors. In Figure 2, sites 3,4,5,7, and 8 are probably internal sites.
2) $s_n = 2$ (left terminal site). A left terminal site has one neighboring site on its right side. In Figure 2, sites 1 and 2 are probably left terminal sites.
3) $s_n = 3$ (right terminal site). A right terminal site has one neighboring site on its left side. In Figure 2, sites 6 and 10 are probably right terminal sites.
4) $s_n = 4$ (Noise). Noise sites are independent entities, so there is no statistical relationship between the amplitude, frequency, or frequency rate of two neighboring noise states. In Figure 2, site 9 is probably a noise site.

Thus, a site can have one of just four possible states. A neighbor site can also be in one of these four states, but can also be related or unrelated.

### *G. Expression for $p(K, J|\mathbf{X})$.*

We now develop an approximate expression for $p(K, J|\mathbf{X})$. Let $F^i_{n,m}$ be a measure of fit from site $n$ to site $m$ given site $n$ is in state $i$. This measure is based on the state variable at state $n$ and the measurement (not the state variable) at site $m$. We could use, for example

$$F^i_{n,m} = N(\mathbf{x}_m - \mathbf{F}\boldsymbol{\theta}^i_n, \mathbf{F}'\mathbf{R}^i_n\mathbf{F} + \boldsymbol{\Sigma} + \mathbf{C}), \tag{4}$$

where $\{\boldsymbol{\theta}^i_n, \mathbf{R}^i_n\}$ are the estimated state variable and covariance of the at site $n$ assuming it is in discrete state $i$, i.e.

$$p(\boldsymbol{\theta}_n|s_n = i, \mathbf{X}) = N(\boldsymbol{\theta}_n - \boldsymbol{\theta}^i_n, \mathbf{R}^i_n).$$

Recall that $\{K, J\}$ designate which neighbors are connected (if any), their states, and the states of the non-connected neighbors. To arrive at $p(K, J|\mathbf{X})$, we arrange all the neighbors of site $n$ into left neighbors and right neighbors. Let left neighbors be designated as sites $A, B, C \ldots$, and

right neighbors be designated as sites $D, E, F \ldots$. Let the first right and first left neighbor, sites $A$ and $D$, be reserved for connected neighbors if there are any. So, if $K$ specifies any connected neighbors, they must be site $A$ and/or $D$. Let $\{c_A, c_B, c_C, ...\}$ , $\{c_D, c_E, c_F, ...\}$ represent the corresponding associations taking values of 0 (not connected) or 1 (connected). Clearly, the only possibilities are $\{c_A, c_B, c_C, ...\} = \{1, 0, 0, ...\}$ or $\{0, 0, 0, ...\}$ , and $\{c_D, c_E, c_F, ...\} = \{1, 0, 0, ...\}$ or $\{0, 0, 0, ...\}$ . We use the approximation

$$\begin{aligned} p(K, J|\mathbf{X}) &= p(s_A, c_A)\, p(s_B, c_B|c_A)\, p(s_C, c_C|c_A, c_B) \\ &\quad \cdot p(s_D, c_D)\, p(s_E, c_E|c_D)\, p(s_F, c_F|c_D, c_E). \end{aligned}$$

The notation is highly symbolic. For example, $p(s_A, c_A)$ is the joint probability that neighbor site $A$ has state $s_A$ and connection state $c_A$. We make use of the approximation that if $A, B$ are left neighbors competing for site $n$, $p(s_B, c_B)$ depends only on whether site $A$ is connected to site $n$ and not on which state it has. Thus,

$$p(s_A, c_A, s_B, c_B) = p(s_A, c_A)\, p(s_B, c_B|c_A).$$

Furthermore, our expectations at site $C$ are not changed by knowing the states at sites $A$ or $B$, once we specify the connections $c_A, c_B$ thus,

$$\begin{aligned} p(s_A, c_A, s_B, c_B, s_C, c_C) &= p(s_A, c_A)\, p(s_B, c_B|c_A) \\ &\quad \cdot p(s_C, c_C|c_A, c_B). \end{aligned}$$

We assume also that our expectations of right neighbors are not influenced by knowledge of left neighbor states or connections. This is clearly not true, but gives us a workable approximation. So, we treat the left neighbors independently in the same way as the right neighbors. Thus,

$$\begin{aligned} p(s_A, c_A, s_B, c_B, s_C, c_C, s_D, c_D, s_E, c_E, s_F, c_F) &= p(s_A, c_A) \\ &\quad \cdot p(s_B, c_B|c_A)\, p(s_C, c_C|c_A, c_B)\, p(s_D, c_D)\, p(s_E, c_E|c_D) \\ &\quad \cdot p(s_F, c_F|c_D, c_E). \end{aligned}$$

The functions $p(s_A, c_A)$ and $p(s_D, c_D)$ are unique because they involve the sites that may be connected to site $n$. Consider the function $p(s_A, c_A)$ given by

| $s_A$ | $c_A$ | $p(s_A, c_A)$ |
|---|---|---|
| 1 | 1 | $p(s_A)\ \alpha(A, n, s_A)$ |
| 2 | 1 | $p(s_A)\ \alpha(A, n, s_A)$ |
| 3 | 1 | 0 |
| 4 | 1 | 0 |
| 1 | 0 | $p(s_A)\ \beta(A, n, s_A)$ |
| 2 | 0 | $p(s_A)$ |
| 3 | 0 | 0 |
| 4 | 0 | 0 |

where

$$\alpha(A, n, s_A) = \frac{F^{s_A}_{A,n}}{\sum_k F^{s_A}_{A,k}},$$

where $k$ runs over all the right neighbors of left neighbor site $A$ which must include site $n$. Clearly $\alpha(A, n, s_A)$ is an approximation of the probability that left neighbor $A$ connects to site $n$ given it is in state $s_A$ and it *does* connect to a right neighbor. Also,

$$\beta(A, n, s_A) = \frac{\sum_{j \neq n} F^{s_A}_{A,j}}{\sum_k F^{s_A}_{A,k}},$$

which is an approximation of the probability that left neighbor $A$ connects to a right neighbor *other than site* $n$ given it is in state $s_A$ and it *does* connect to a right neighbor. Logically, $p(s_A = \{1 \text{ or } 2\}, s_A = 0) = 0$ when $n$ is the only right neighbor of left neighbor site $A$. Tabulating $p(s_B, c_B|c_A)$,

| $s_B$ | $c_B$ | $c_A$ | $p(s_B, c_B|c_A)$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | 1 | 1 | 0 |
| 1 | 0 | 1 | $p(s_B)\ \beta(B, n, s_B)$ |
| 2 | 0 | 1 | $p(s_B)\ \beta(B, n, s_B)$ |
| 3 | 0 | 1 | $p(s_B)$ |
| 4 | 0 | 1 | $p(s_B)$ |
| 1 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 |
| 1 | 0 | 0 | $p(s_B)\ \beta(B, n, s_B)$ |
| 2 | 0 | 0 | $p(s_B)\ \beta(B, n, s_B)$ |
| 3 | 0 | 0 | $p(s_B)$ |
| 4 | 0 | 0 | $p(s_B)$ |

where $p(s_B)$ is obtained from the initial state probabilities $Q_{B,s_B}$, and $\beta(B, n, s_B)$ represents the probability that left neighbor $B$ connects to a right neighbor other than site $n$ given it is in state $s_B$ and given it does connect to a right neighbor. Note that we gave set $p(s_B, c_B|c_A) = 0$ whenever $c_B = 1$ because, as we stated before, we have ordered the left neighbors so that if any of them connect to site $n$, it is site $A$. We assume that given $s_B$ is a non-connected state (3 or 4), it does not depend on $c_A$, although in principle knowing $c_A = 1$ should increase the likelihood that $B$ is in a non-connected state since it is deprived of a potential right neighbor. But the complications are many and payoff is small. The expression for $p(s_C, c_C|c_A, c_B)$ must enumerate all combinations of $c_A, c_B$, but thanks to the fact that we ordered the neighbors so that only site $A$ can connect, we find that the $p(s_C, c_C|c_A, c_B)$ may be evaluated using the table for $p(s_B, c_B|c_A)$ by substituting $C, s_C, c_C$ for $B, s_B, c_B$ and multiplying by the function $\delta[c_B = 0]$ which is one whenever $c_B = 0$ and zero when $c_B = 1$. Additional left neighbors are accomodated in a similar manner. For right neighbors, we repeat the tables of $p(s_A, c_A)$ and $p(s_B, c_B|c_A)$ substituting the right neighbor sites for the left neighbor sites.

### *H. Algorithm Summary*

Given:

1) Measurements $\mathbf{x}_n$ for $1 \leq n \leq L$.

2) Measurements error covariance $\mathbf{C}$. You may use site-dependent (i.e. SNR dependent) error covariance $\mathbf{C}_n$.
3) System dynamical model. Let $\boldsymbol{\theta}$ be the state variable at a site corresponding to one of the $P$ possible states. Let $\boldsymbol{\phi}$ be the collection of neighbor state variables (either a single left or right neighbor, or both left and right neighbors) corresponding to some choice of neighbor state combination. We assume that $\boldsymbol{\theta}$ is determined from $\boldsymbol{\phi}$ according to

$$\boldsymbol{\theta} = \mathbf{F}\boldsymbol{\phi} + \mathbf{w},$$

where $\mathbf{w}$ is a Gaussian RV with covariance $\boldsymbol{\Sigma}$. Note that $\mathbf{F}$ and $\boldsymbol{\Sigma}$ will depend on the whether there is a single left neighbor, a right neighbor, or both left and right neighbors.

Initialization:

1) For $1 \le n \le L$, $1 \le i \le P$, initialize $Q_{n,i}$, and the mean and covariances of $p_n(\boldsymbol{\theta}|i, \mathbf{X})$, denoted by $\boldsymbol{\theta}_n^i$, $\mathbf{R}_n^i$.

Algorithm:

1) For $1 \le n \le L$, $1 \le m \le L$, $1 \le i \le P$ compute the measure of fit $F_{n,m}^i$ according to (4) (note this is a sparse matrix as only neighbor pairs need be considered).
2) For each $1 \le n \le L$,
   a) For each $1 \le i \le P$,
      i) Initialize $\hat{p}_n^i = 0$, $\hat{\boldsymbol{\theta}}_n^i = 0$, $\hat{\mathbf{R}}_n^i = 0$,
      ii) Determine the list of all possible choices for associated neighbors of site $n$, denoted by $A$, and their states, denoted by $J_A$, which support state $i$.
      iii) For each combination $K = \{A, J_A\}$,
         A) Compute $\mathbf{G}_K$, $\mathbf{u}_K$ according to (1), (2).
         B) For each combination $J$ of states of the non-associated neighbors, compute $p(K, J|\mathbf{X})$ according to Section II-G.
         C) Let $w_K = \sum_J p(K, J|\mathbf{X})$.
         D) Accumulate $\hat{\boldsymbol{\theta}}_n^i = \hat{\boldsymbol{\theta}}_n^i + w_K \mathbf{u}_K$
         E) Accumulate $\hat{\mathbf{R}}_n^i = \hat{\mathbf{R}}_n^i + w_K(\mathbf{G}_K + \mathbf{u}_K \mathbf{u}_K')$
      iv) Normalize $\hat{\boldsymbol{\theta}}_n^i = \hat{\boldsymbol{\theta}}_n^i / \sum_K w_K$.
      v) Normalize $\hat{\mathbf{R}}_n^i = \hat{\mathbf{R}}_n^i / \sum_K w_K - \hat{\boldsymbol{\theta}}_n^i \hat{\boldsymbol{\theta}}_n^{i\prime}$.
      vi) Accumulate $\hat{p}_n^i = \hat{p}_n^i + \sum_K w_K$.
3) Normalize $\hat{p}_i = \frac{\hat{p}_i}{\sum_i \hat{p}_i}$.
4) Complete iteration: For $1 \le n \le L$,
   a) Set $Q_{n,i} = \hat{p}_i$.
   b) Set $\boldsymbol{\theta}_n^i = \hat{\boldsymbol{\theta}}_i$.
   c) Set $\mathbf{R}_n^i = \hat{\mathbf{R}}_i$.
5) Go to top.

### *I. Interpretation of Results*

When the algorithm converges, it is necessary to interpret the results. What one has after convergence are estimates of the discrete state probabilities of each site as well as a continous-valued state variable estimates and covariance estimates for each site/state combination. One also has the fit function $F_{n,m}^i$ calculated from the state variables and covariances. It is necessary to search the graph for tracks. One way to search the graph is to locate any sites with high probability of being a left terminal site (state 2), then find the most likely right neighbor by maximizing $F_{n,m}^i$ over the indexes $m$ corresponding to right neighbors, then continuing to the right until a probable right terminal site is found. This method can fail if, for example, there are multiple candidates for left terminal site and the probability is shared among them, reducing the probability of state 2 in each candidate. Thus, it is a good idea to also search starting at the right, or in the middle. We have found the best approach to be to look for high probabilities of state 1 (internal site), then follow the trail both left and right until a site either has no further neighbor or it has a high probability of being a noise site (state 4).

## III. Simulation

Simulated data was created by adding signal to a 30 by 30 grid of exponentially-distributed noise with mean 2.0. The target is injected by adding an interpolated function using a Gaussian shape of variance 0.5 (in the vertical direction) having a maximum value of 2.5. Figure 3 shows a sample of the simulated data and has rejected the downward branch to sites 33, 42, 50. On the right panel, the true target track is superimposed. On the left panel, the eye has a difficult time seeing the target without the guide. In figure 4, the output detections of the Hough transform are shown. For each detection, the detection (site) number is printed. The site center is indicated by a dot and the slope is represnted by a line. Notice that the signal track is clearly visible to the eye. However, there are other potential tracks as well as interfering "branches" that may cause incorrect results. In particular, as the target track proceeds to the right from site 20 to site 26, it could proceed to site 34,44,52, or it branch down to site 33, 42, and 50. Figure 5 and 6 show the result after one and two iterations of the MRF algorithm. The target track is beginning to "crystalize". Those sites in state 1 are shown as blue lines centered on the time location of the site (site 42 for example). Those sites in states 2 are shown as green lines extending to the right (site 33 for example). Those sites in states 3 are shown as red lines extending to the left (site 50 for example). Noise sites are shown as magenta circles. Already after 2 iterations, the algorithm has apparently settled on site 6 at the left end. On the right end, there is a possibility that the target ends at either sites 61 or 63. Furthermore, site 63 has another potential left neighbor, site 54, which is unrelated to the target. In figure 7, taken after 50 iterations, we see that the target track has straightened out considerably. And, the algorithm has provided a diplomatic solution to the right-end dispute. It decided that the track ends at site 52 after all, leaving sites 54 and 63 free to embrace each other, and letting site 61 die. This is apparently because it would require too abrupt a maneuver to curve down to site 61, and site 63 preferred to go with site 52 anyway.

In trials, there was a 70 percent probability of success, measured as the existance of a track with the correct slope and length, within some bounds.

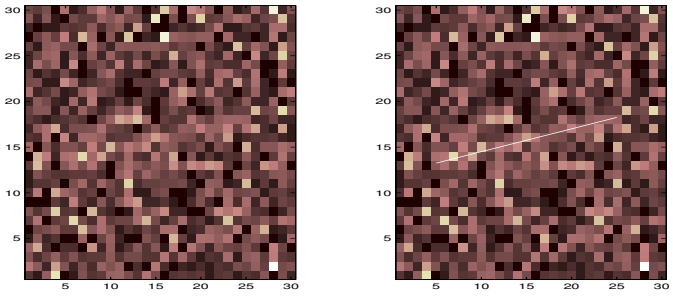

Fig. 3. Simulated Data

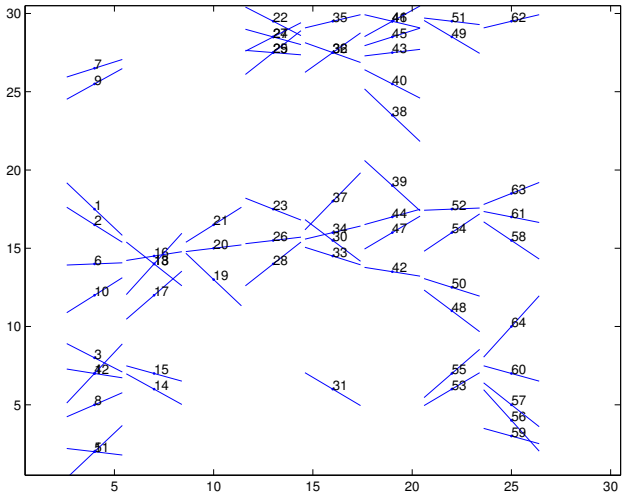

Fig. 4. Hough Transform Outputs

### A. Conclusion

We have posed the target tracking problem as Markov Random Field (MRF). However, rather than working with the Gibbs field representation of an MRF, we have based the algorithm directly on the probabilistic expression of the Markov property, the conditional PDF of a site's state variable given the neightbor state variables and the site measurement. We have used a dynamical model akin to a Kalman filter to model the target dynamics. The algorithm is iterative. By using the Gaussian assumption throughout, we obtain a simplifed and efficient algorithm. We demonstrated the algorithm using synthetic data.

## REFERENCES


[1] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, p. 721741, 1984.

[2] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. M. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization," *Journal of Machine Learning Research*, vol. 1, pp. 49–75, 2000.

[3] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.


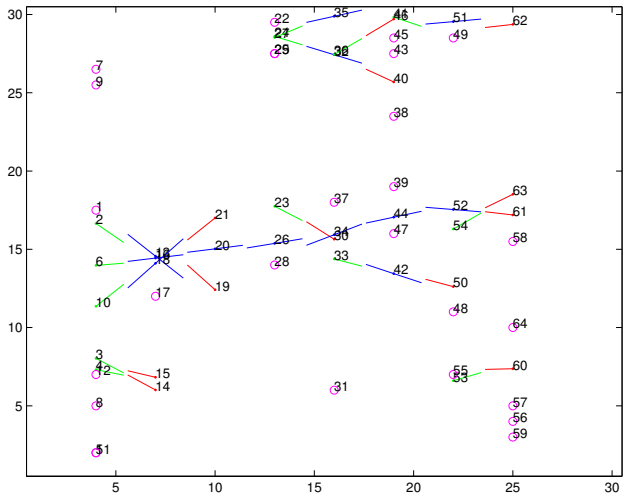

Fig. 5. After iteration 1

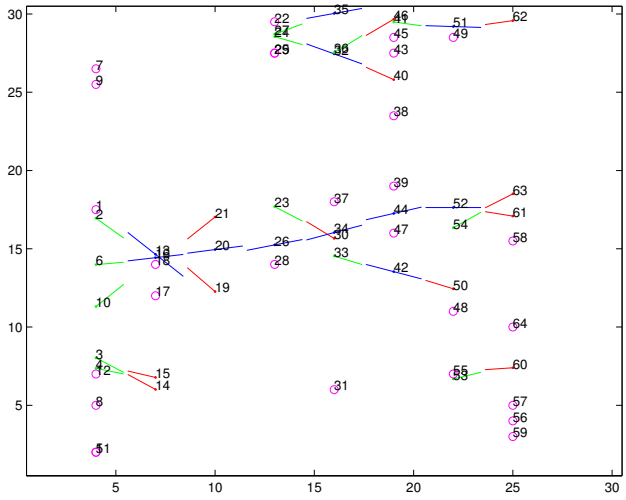

Fig. 6. After iteration 2

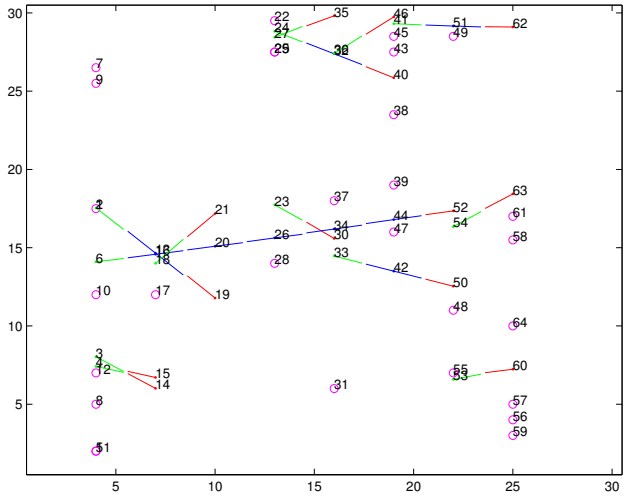

Fig. 7. After Iteration 50